

- Sélection
- Recherche
- Ordonner les réponses

- Fonctions de groupe
- Sous-requêtes
- Opérateurs de l'algèbre relationnelle

# Fonctions de groupe

Comment calculer ...

- Le nombre d'avions pour chaque compagnie?
- Les compagnies qui ont plus de trois avions ?
- La moyenne des heures de vols des pilotes ?

# Les fonctions de groupe

---

- Des fonctions de calcul
- Le regroupement: Clause GROUP BY
- Fonctions liées au GROUP BY
- Clause HAVING

- Sélection
- Recherche
- Ordonner les réponses

- Fonctions de groupe
- Sous-requêtes
- Opérateurs de l'algèbre relationnelle

# Fonctions de groupe

Les fonctions de groupe calculent les résultats à partir d'une collection de valeurs.

| Fonction                  | Description   |
|---------------------------|---|
| COUNT (*)                 | Comptage des lignes ( <b>prend en compte les valeurs NULL</b> )               |
| COUNT ([DISTINCT] exp)    | Comptage des valeurs ( <b>les valeurs NULL ne sont pas prises en compte</b> ) |
| MAX ([DISTINCT] exp)      | Maximum de <i>exp</i> (nombre, date, chaîne)                                  |
| MIN ([DISTINCT] exp)      | Minimum de <i>exp</i> (nombre, date, chaîne)                                  |
| SUM ([DISTINCT] exp)      | Somme de <i>exp</i> (valeurs)   |
| AVG ([DISTINCT] exp)      | Moyenne de <i>exp</i> ( <i>nombre</i> )                                       |
| STDDEV ([DISTINCT] exp)   | Écart type de <i>exp</i> ( <i>nombre</i> )                                    |
| VARIANCE ([DISTINCT] exp) | Variance de <i>exp</i> ( <i>exp</i> )   |
| Et bien d'autres...       |   |

- Sélection
- Recherche
- Ordonner les réponses

- Fonctions de groupe
- Sous-requêtes
- Opérateurs de l'algèbre relationnelle

# Fonctions de groupe

## Exemples :

- Pour connaître le nombre de vols :

```
SELECT count (*) NbreVols FROM VOL ;
```

NbreVols

60

- Pour connaître le nombre de pilotes qui ont réalisé des vols:

**Les valeurs null et doublons ne sont pas comptées**

```
SELECT count (distinct numP) NbrePilote  
FROM Vol ;
```

NbrePilote

13

```
SELECT count (numP) NbrePilote  
FROM Vol ;
```

NbrePilote

57

**Les valeurs null ne sont pas comptées**

- Sélection
- Recherche
- Ordonner les réponses

- Fonctions de groupe
- Sous-requêtes
- Opérateurs de l'algèbre relationnelle

# Fonctions de groupe

- Déterminer la moyenne de vol des pilotes de la compagnie 'AF'

```
SELECT AVG(nbhvol)
FROM Pilote
WHERE comp='AF' ;
```

|              |
|--------------|
| AVG (nbhvol) |
| 225          |

- Déterminer le nombre d'heures de vol le plus élevé et le nombre de pilotes

```
SELECT MAX(nbhvol), COUNT (*) AS NbrePilotes
FROM Pilote;
```

| MAX(nbhvol) | NbrePilotes |
|-------------|-------------|
| 2450        | 7           |

# Regroupement de lignes

- **GROUP BY:** s'emploie avec l'instruction SELECT pour classer des données identiques par groupes.

```
SELECT COLONNE1, COLONNE2
FROM <liste_de_tables>
WHERE condition
GROUP BY COLONNE1, COLONNE2;
```

- A placer après le WHERE et avant le ORDER BY;
- **REMARQUE:** les colonnes présentes dans le **SELECT doivent apparaître dans le GROUP BY.** Seules les fonctions ou expressions peuvent exister en plus dans le SELECT.
- **Attention on ne peut pas avoir de colonnes simples et des fonctions de calcul dans une requête sans avoir une fonction GROUP BY;**

- Sélection
- Recherche
- Ordonner les réponses

- Fonctions de groupe
- Sous-requêtes
- Opérateurs de l'algèbre relationnelle

# Regroupement de lignes

## Exemple :

Un groupement sur la colonne comp de la table PILOTE

| numP | nom              | nbhvol | comp |
|------|------------------|--------|------|
| 1200 | Gratien Viel     | 450    | AF   |
| 300  | Didier Donsez    | 0      | AF   |
| 4001 | Richard Grin     | 1000   | SING |
| 6000 | Francoise Tort   | 500    | CAST |
| 7006 | Placide Fresnais | 2450   | SING |

```
SELECT
    comp
  FROM PILOTE
  GROUP BY comp;
```

| numP | nom            | nbhvol | comp |
|------|----------------|--------|------|
| 1200 | Gratien Vie    |        | AF   |
| 300  | Didier Dons    |        | AF   |
| 4001 | Richard Grin   |        | SING |
| 7006 | Placide Fres   |        | SING |
| 6000 | Francoise Tort | 500    | CAST |

# Regroupement de lignes

---

- La requête ne renvoie **qu'un seul n-uplet par groupe**
- Le **SELECT** et le **ORDER BY** ne peuvent utiliser que des **attributs présents dans le GROUP BY**
  - Dans un groupe, un attribut de regroupement a la même valeur pour tous les n-uplets, donc utilisable directement.
  - Les autres attributs ont des valeurs diverses, donc non utilisables directement: utilisation via des fonctions.

- Sélection
- Recherche
- Ordonner les réponses

- Fonctions de groupe
- Sous-requêtes
- Opérateurs de l'algèbre relationnelle

# Regroupement de lignes

## Exemple :

Pour connaître le nombre d'avions affectés à chaque ville d'affectation d'un avion :

```
SELECT Loc, count(*) NbreAvions
FROM Avion
GROUP BY Loc ;
```

Avions triés sur loc

| numAv | loc       |
|-------|-----------|
| 820   | Ajaccio   |
| 715   | Ajaccio   |
| ..... | .....     |
| 720   | Marseille |
| 225   | Marseille |
| 456   | Marseille |
| ..... | ....      |
| 531   | Singapour |

Table résultat

| Loc       | nbreAvions |
|-----------|------------|
| Ajaccio   | 2          |
| .....     | .....      |
| Marseille | 3          |
| .....     | ....       |
| Singapour | 1          |



- Sélection
- Recherche
- Ordonner les réponses

- Fonctions de groupe
- Sous-requêtes
- Opérateurs de l'algèbre relationnelle

# Regroupement de lignes sélectionnées

## Exemple :

Pour connaître le nombre d'avions différents utilisés par chaque pilote assurant un vol :

```
SELECT      NumP, count(distinct NumAv) NbreAvions
FROM    Vol
WHERE  NumP is not null
GROUP BY NumP ;
```

- Sélection
- Recherche
- Ordonner les réponses

- Fonctions de groupe
- Sous-requêtes
- Opérateurs de l'algèbre relationnelle

# Regroupement de lignes

Clause **HAVING**: pose de conditions sur chaque regroupement et sélectionne les groupes qui satisfont la condition

```
SELECT COLONNE, COLONNE2
FROM <liste_de_tables>
WHERE condition
GROUP BY COLONNE1, COLONNE2
HAVING condition;
```

- Exécutée entre le GROUP et le ORDER BY

- Sélection
- Recherche
- Ordonner les réponses

- Fonctions de groupe
- Sous-requêtes
- Opérateurs de l'algèbre relationnelle

# Condition sur l'ensemble des lignes

## Exemple :

- Déterminer les compagnies ayant un nombre d'heures total de vols des pilotes qui dépasserait 200

```
SELECT      sum(nbhvol) ,  
comp  
FROM  pilote  
WHERE sum(nbHVol)>200  
Group By comp;
```

FAUX

| COMP  | SUM(NBHVOL) |
|-------|-------------|
| ----- | -----       |
| AF    | 450         |
| AC    | 400         |
| SING  | 3650        |

```
SELECT      sum(nbhvol) , comp  
FROM  pilote  
Group By comp  
HAVING sum(nbhvol)>200;
```

- Sélection
- Recherche
- Ordonner les réponses

- Fonctions de groupe
- Sous-requêtes
- Opérateurs de l'algèbre relationnelle

# Sélection des groupes de lignes

## Exemple :

Pour connaître les pilotes qui conduisent au moins deux avions différents :

```
SELECT      numP
FROM        Vol
WHERE       numP is not null
GROUP BY    numP
HAVING      count(DISTINCT NumAv) >= 2
;
```

| NUMVOL | NUMP | NUMAV |
|--------|------|-------|
| 1      | 300  | 200   |
| 2      | 4001 | 300   |
| 490    | 7006 | 800   |
| 500    | 7006 | 300   |



NUMP

7006

- Sélection
- Recherche
- Ordonner les réponses

- Fonctions de groupe
- Sous-requêtes
- Opérateurs de l'algèbre relationnelle

# En résumé

---

- Clause GROUP BY pour créer des groupes

- Fonctions liées au regroupement

- Clause HAVING pour filtrer les groupes

- Syntaxe:

```
SELECT COLONNE1, COLONNE2
FROM <liste_de_tables>
WHERE condition
GROUP BY COLONNE1, COLONNE2
HAVING condition;
```